

---

# オブジェクト指向レポート

## オブジェクト指向の3大要素を活用する

竹村 太希 - 平成29年12月3日

---

## 各クラスのソースコード

ここに各クラスのソースコードを記載する。

### ・ Carクラス

```
public abstract class Car {
    protected int speed;
    protected int tire;
    protected int door;
    protected int gas;
    protected int nenpi;

    Car(){
        this(100, 5, 4);
    }
    Car(int speed, int door, int tire){
        this.speed = speed;
        this.door = door;
        this.tire = tire;
        System.out.println("車を製造");
    }

    public void setSpeed(int speed){
        if(speed > 180 || speed < 40){
            System.out.println("異常な速度が指定されました。速度を50キロに設定します。");
            speed = 50;
        }
        this.speed = speed;
        System.out.println("時速:" + this.speed);
    }

    public void setTire(int tire){
        if(tire > 5 || tire < 3){
            System.out.println("異常なタイヤの数が指定されました。タイヤを4つに設定します。");
            tire = 4;
        }
        this.tire = tire;
        System.out.println("タイヤ:" + this.tire);
    }

    public void setDoor(int door){
        if (door > 5 || door < 1) {
            System.out.println("異常なドアの数が指定されました。タイヤを4つに設定します。");
        }
        this.door = door;
        System.out.println("ドア" + this.door);
    }
    public abstract void setGas(int gas);
    public abstract void setNenpi(int nenpi);
}
```

---

```

public void drive(){
    if(0 >= getGas()){
        System.out.println("ガス欠です");
    } else {
        // これからのスタンダードはハイブリッドや電気自動車であると考えられ、本プログラムは将来を見据えた設計とした
        System.out.print("キーン☆ c");
        setGas(getGas() - getNempi());
    }
}

public int getSpeed(){
    return speed;
}

public int getDoor(){
    return door;
}

public int getGas(){
    return gas;
}

public int getNempi(){
    return nenpi;
}

public int getTire(){
    return tire;
}
}

```

## ・ Truckクラス

```

public class Truck extends Car{
    private int nimotsu;
    Truck(){
        System.out.println("トラックを製造");
    }

    public void setNimotsu(int nimotsu){
        if (nimotsu > 300) {
            System.out.println("異常な荷物の数が指定されました。50個に設定します。");
            nimotsu = 50;
        }
        this.nimotsu = nimotsu;
        System.out.println("荷物の数:" + this.nimotsu);
    }
}

```

---

```

@Override
public void setSpeed(int speed){
    if(speed > 120 || speed < 40){
        System.out.println("異常な速度が指定されました。速度を80キロに設定します。");
        speed = 80;
    }
    this.speed = speed;
    System.out.println("時速:" + this.speed);
}

public void setGas(int gas){
    if (gas > 600) {
        System.out.println("異常な積載燃料が指定されました。積載燃料を30Lに設定します。");
        gas = 30;
    }
    this.gas = gas;
    System.out.println("現在の燃料:" + this.gas);
}

public void setNenpi(int nenpi){
    if (nenpi > 10 || nenpi < 2) {
        System.out.println("異常な燃費が指定されました。燃費を4km/Lに設定します。");
        nenpi = 4;
    }
    this.nenpi = nenpi;
    System.out.println("燃費:" + this.nenpi);
}

public int getNimotsu(){
    return nimotsu;
}
}

```

#### ・ Sirenインターフェース

```

public interface Siren {
    void callSiren();
}

```

#### ・ Ambulanceクラス

```

public class Ambulance extends Car implements Siren {
    Ambulance(){
        System.out.println("救急車を製造");
    }

    public void callSiren(){
        System.out.println("ピーポーピーポー");
    }

    public void setGas(int gas){

```

---

```
if (gas > 120) {
    System.out.println("異常な積載燃料が指定されました。積載燃料を80Lに設定します。");
    gas = 80;
}
this.gas = gas;
System.out.println("現在の燃料:" + this.gas);
}

public void setNenpi(int nenpi){
    if (nenpi > 10 || nenpi < 2) {
        System.out.println("異常な燃費が指定されました。燃費を5km/Lに設定します。");
        nenpi = 5;
    }
    this.nenpi = nenpi;
    System.out.println("燃費:" + this.nenpi);
}
}
```

## ・ PatrolCarクラス

```
class PatrolCar extends Car implements Siren {
    PatrolCar(){
        System.out.println("パトロールカーを製造");
    }

    public void callSiren(){
        System.out.println("ファンファンファン");
    }

    @Override
    public void setSpeed(int speed){
        //最新の覆面パトカーは260km出るような
        if(speed > 260 || speed < 40){
            System.out.println("異常な速度が指定されました。速度を80キロに設定します。");
            speed = 80;
        }
        this.speed = speed;
        System.out.println("時速:" + this.speed);
    }

    public void setGas(int gas){
        if (gas > 100) {
            System.out.println("異常な積載燃料が指定されました。積載燃料を50Lに設定します。");
            gas = 50;
        }
        this.gas = gas;
        System.out.println("現在の燃料:" + this.gas);
    }

    public void setNenpi(int nenpi){
```

---

```
if (nenpi > 30 || nenpi < 10) {
    System.out.println("異常な燃費が指定されました。燃費を15km/Lに設定します。");
    nenpi = 15;
}
this.nenpi = nenpi;
System.out.println("燃費:" + this.nenpi);
}
}
```

## ・ Mainクラス

```
public class Main{
    public static void main(String args[]){
        System.out.println("===トラック===");

        Truck hino2ton = new Truck();
        hino2ton.setGas(200);
        hino2ton.setDoor(2);
        hino2ton.setTire(6);
        hino2ton.setNenpi(4);
        hino2ton.setSpeed(100);
        hino2ton.setNimotsu(50);
        hino2ton.drive();
        hino2ton.drive();
        hino2ton.drive();
        hino2ton.drive();
        System.out.println("===救急車===");

        Ambulance kyukyu = new Ambulance();
        kyukyu.setDoor(3);
        kyukyu.setSpeed(500);
        kyukyu.setTire(4);
        kyukyu.setGas(100);
        kyukyu.setNenpi(5);
        kyukyu.callSiren();
        System.out.println("===パトカー===");

        PatrolCar patrol = new PatrolCar();
        patrol.setDoor(4);
        patrol.setTire(4);
        patrol.setSpeed(200);
        patrol.setGas(40);
        patrol.setNenpi(20);
        patrol.callSiren();
        patrol.drive();
        patrol.callSiren();
        patrol.drive();
        patrol.callSiren();
        patrol.drive();
        patrol.callSiren();
        patrol.drive();
    }
}
```

---

## 実行結果

===トラック===

車を製造

トラックを製造

現在の燃料:200

ドア2

異常なタイヤの数が指定されました。タイヤを4つに設定します。

タイヤ:4

燃費:4

時速:100

荷物の数:50

キーン☆現在の燃料:196

キーン☆現在の燃料:192

キーン☆現在の燃料:188

キーン☆現在の燃料:184

===救急車===

車を製造

救急車を製造

ドア3

異常な速度が指定されました。速度を50キロに設定します。

時速:50

タイヤ:4

現在の燃料:100

燃費:5

ピーポーピーポー

===パトカー===

車を製造

パトロールカーを製造

ドア4

タイヤ:4

時速:200

現在の燃料:40

---

燃費:20

ファンファンファン

キーン☆現在の燃料:20

ファンファンファン

キーン☆現在の燃料:0

ファンファンファン

ガス欠です

ファンファンファン

ガス欠です